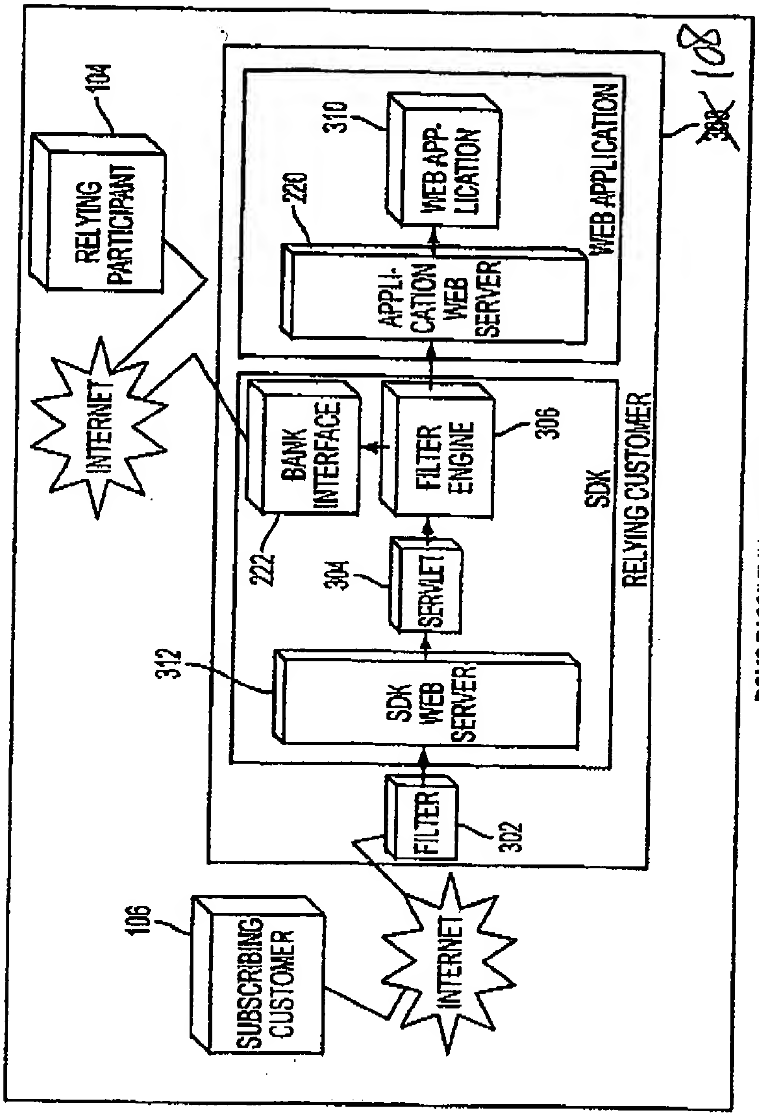
REPLACEMENT SHEET
Serial No.: 09/657,604
Filed: September 8, 2000
Inventors: Jackson Brandenburg et al.
Sheet 3 of 19



DSMS PASSIVE INTEGRATION

FIG. 3

REPLACEMENT SHEET
Serial No.: 09/657,604
Filed: September 8, 2000
Inventors: Jackson Brandenburg et al.
Sheet 5 of 19

	Steps
	Load Servlet properties from the properties file
504	Read data from the HTTP request
506	Create a hash table (name, value pairs) with parameters for the Filter Engine including HTTP headers, Content type, client IP address, HTTP method (GET and SET) and the actual data in the request
508	Identify if the data has been signed. If not signed, call Filter Engine with the hash table
512	If signed, URL decode the PKCS#7 message received from the Plug-In and insert it into the hash table
510	Call the Filter Engine with the hash table
514	Process the return value from the Filter Engine
516	If the return value from the Filter Engine indicates that the web application has been called, then display the next page
518	If the return value from the Filter Engine indicates that the page needs to be signed, the state of the Filter Engine is stored in a cookie and the page with the Plug-In is displayed
520	If the return value from the Filter Engine indicates that the Client Certificate is GOOD, then change the State and send a request to Filter Engine to retrieve the next page.
522	For all other values or exceptions, display error page to the client.

Fig. 5

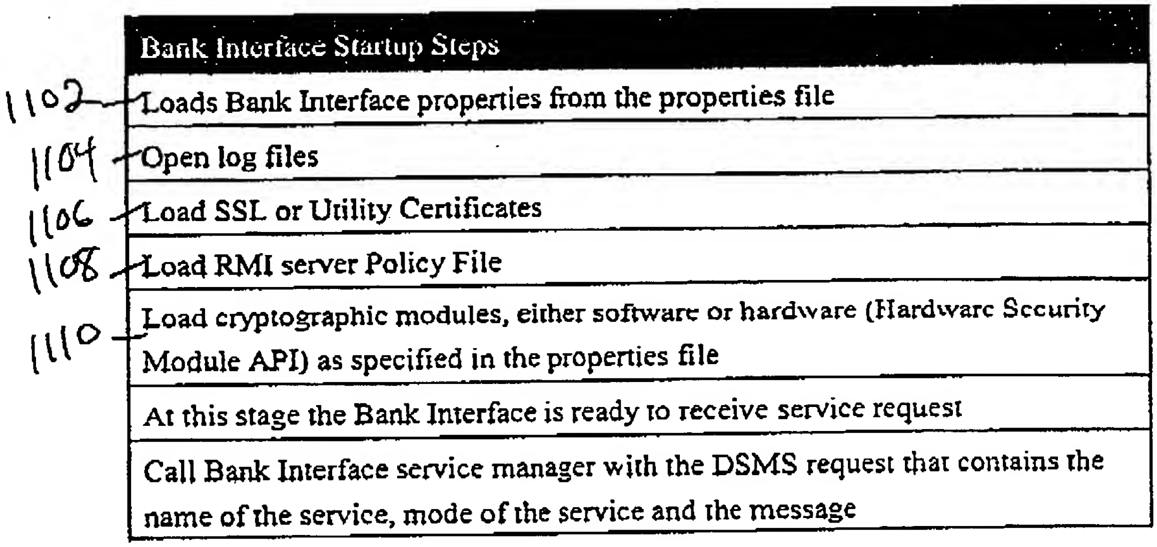
	Filter Engine Startup Steps
802	Loads Filter Engine properties from the properties file
804	Open log files
806	Load SSL or Utility Certificates
808	Load RMI server Policy File
810	Load Rules files into the memory
812	Validate Rules to verify correct formatting
	The Filter Engine Interface is now ready to receive requests

Fig. 8

	Filter Engine Processing Steps
902	Receives HTTP Request data and the State from the Servlet
904	If the State passed from the Servlet is FE_NEW_REQUEST, the Filter Engine compares the request against the signing rules and determines whether the request has to be signed or not. It builds the Return Object specified in the FE_NEW_REQUEST State.
906	If the State passed in from the Servlet is FE_SIGNED_DATA, then it calls the Bank Interface to check the status of the Certificate. After interacting with the Identrus network, the Bank Interface returns the status. The status and the data in the CMS message are put into a Return Object and sent to the Servlet
908	If the State passed from the Servlet is FE_REQUEST_CHECKED, indicating the final stage of a signed transaction, the Web Application is called. The original page is retrieved from the Web Application and its content is returned to the Servlet in a Return Object
	Log all signed request to the event log and all errors to the error log
	All exceptions are returned to the Servlet as a part of the Return Object

Fig. 9

REPLACEMENT SHEET
Serial No.: 09/657,604
Filed: September 8, 2000
Inventors: Jackson Brandenburg et al.
Sheet 10 of 19



Bank Interface Startup Steps

1102 Loads Bank Interface properties from the properties file

1104 Open log files

1106 Load SSL or Utility Certificates

1108 Load RMI server Policy File

1110 Load cryptographic modules, either software or hardware (Hardware Security Module API) as specified in the properties file

At this stage the Bank Interface is ready to receive service request

Call Bank Interface service manager with the DSMS request that contains the name of the service, mode of the service and the message

Fig. 11

REPLACEMENT SHEET
Serial No.: 09/657,604
Filed: September 8, 2000
Inventors: Jackson Brandenburg et al.
Sheet 11 of 19

	Steps
1202	Retrieve Relying Customer and Root Certificate from the server
1204	Retrieve Subscribing Customer and Issuing Participant's Certificate from the CMS (Cryptographic Message Syntax) also referred as PKCS#7.
1206	Verify signature on the CMS message
1208	Verify signature on the Subscribing Customer's Certificate using the Issuing Participant's Certificate
1210	Verify signature on the Issuing Participant's Certificate using the Identrus Root Certificate that belongs to the Relying Participant
1212	The Validity period is then checked on the two Certificates received against the current date
1214	Retrieve the OCSP responder's URL from the Relying Customer's certificate
1216	Create an OCSP request for the Subscribing Customer's Certificate signed by the Relying Customer. All OCSP requests contain a Service Locator Extension, which is set by the Authority Information Access (AIA) extension defined in the certificate
1218	Log the OCSP request to the transaction log
1220	Create HTTP(S) connection to the OCSP responder and send the OCSP request.
1222	Receive OCSP response from the responder and verify the signature using the OCSP Responder's Certificate
1224	Get the status of the certificate from the Response
1226	Repeat steps 8 through 11 for the Issuing Participant and the Relying Participant's OCSP Responder's certificate
1228	Log the OCSP response to the transaction log
1230	If the status of all the responses are GOOD return GOOD, else return the status
1232	Log all signed request to the event log and all errors to the error log
	All exceptions are returned to the client as a part of the Return Object

Fig. 12

REPLACEMENT SHEET
Serial No.: 09/657,604
Filed: September 8, 2000
Inventors: Jackson Brandenburg et al.
Sheet 12 of 19

#	Description	Protocol
1301 ~~1~~	User clicks 'Submit' button on HTML Form in Web Browser	HTML UI
1302 ~~2~~	Web Browser posts form data to SDK Web Server	HTTP
1303 ~~3~~	SDK Web Server passes all requests to Servlet.	
1304 ~~4~~	Servlet passes request to Filter Engine.	RMI
1305 ~~5~~	Filter Engine creates a Return-to-Browser URL (as a GET with parameters for data) representing the data of the original POST or GET form posting and returns it along with instructions to get the data signed to the Servlet	RMI
1306 ~~6~~	Servlet builds a response with 1. An Applet tag pointing to the Client Interface Applet OR 2. A call to a browser plug-in and the arguments Return-to-Browser URL and the data to sign	Servlet
1307	SDK Web Server returns the Servlet's response to the Web Browser.	HTTP
1308 ~~8~~	Web Browser displays the HTML Page (requests the Applet if necessary)	HTTP
1309 ~~11~~	Web browser displays Client Interface Applet or activates the plug-in, The arguments are the data to sign and possibly a URL	Browser
1310 ~~12~~	User clicks button in to approve signing of form data.	GUI
1311 ~~13~~	Client Interface (applet or plugin) calls Smart Card API to request that the Smart Card sign an SHA-1 hash of the form data.	Client Interface
1312 ~~15~~	User enters PIN when driver ask for it.	OS Dialog
1313 ~~18~~	Smart Card API returns signed form data to Client Interface.	Client Interface
1314 ~~19~~	Client Interface makes a HTTP connection to the SDK Web Server and submits the signed form data.	HTTP
1315 ~~20~~	SDK Web Server passes request to Servlet	Servlet
1316 ~~21~~	Servlet passes request to Filter Engine.	RMI
1317 ~~22~~	Filter Engine calls Bank Interface with signed data.	RMI

Fig. 13A

REPLACEMENT SHEET
Serial No.: 09/657,604
Filed: September 8, 2000
Inventors. Jackson Brandenburg et al.
Sheet 13 of 19

1318	~~23~~	The Bank Interface calls the Open Card API to request that the HSM sign an SHA-1 hash of the request to the bank.	Java Function Call
1319	~~24~~	Open Card API calls HSM OS Driver	Java Native Call
1320	~~25~~	HSM OS Driver calls HSM to perform signature.	OS-Level Hardware Call
1322	~~26~~	HSM OS Driver returns signed request to Open Card API	Java Native Call
27	~~27~~	Open Card API returns signed request to Bank Interface	Java Function Call
2028	~~28~~	Bank Interface calls the relying party's bank	Warranty/Status Check
2029	~~29~~	Relying party's bank calls the issuing party's bank.	Warranty/OCSP
1330	~~30~~	Issuing party's bank returns a signed response to the relying party's bank.	Warranty/OCSP
1331	~~31~~	Relying party's bank then calls the root.	Warranty/OCSP
1332	~~32~~	Root returns a signed response to the relying party's bank.	Warranty/OCSP
1333	~~33~~	Relying party's bank returns a signed response to the Bank Interface.	Warranty/Status Check
1334	~~34~~	Bank Interface validates the signed data and then records the transaction in the log.	File I/O
1335	~~35~~	Bank Interface validates the signed data and then stores the signed data and the signed response from the relying party's bank into the SDK's database.	JDBC
1336	~~36~~	Bank Interface returns an OK or failure result to Filger Engine	RMI
1337	~~37~~	Filter Engine returns failure result to Servlet or passes on initial request to App Server.	RMI
1338	~~38~~	Servlet builds response indicating failure for SDK Web Server.	Servlet
1339	~~39~~	SDK Web Server returns servlet response to the browser if failure.	HTTP
	45	Web Application's Web Server calls the Web Application	ISA
	46	Web Application generates and returns its response.	ISA
	47	Web Application's Web Server returns the response to the Filter Engine	HTTP

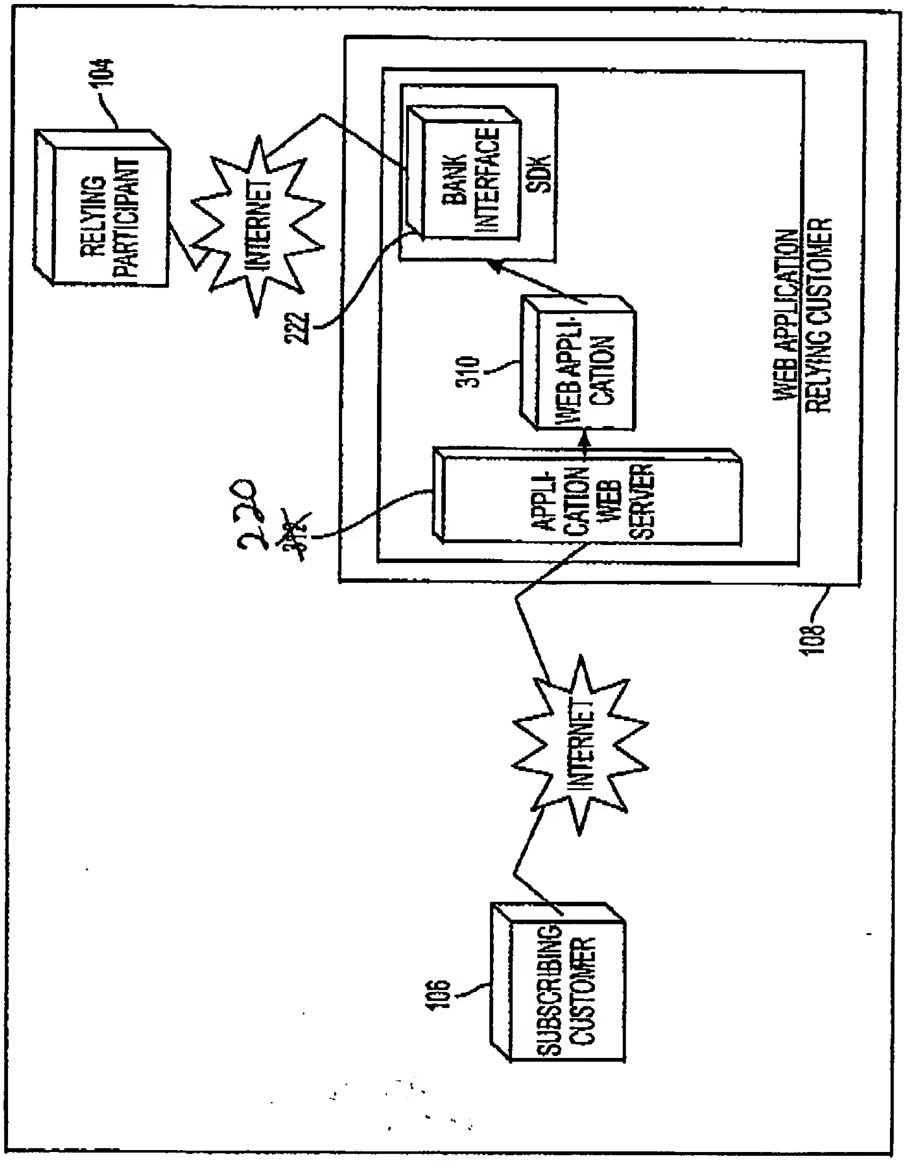
F. 13

REPLACEMENT SHEET
Serial No.: 09/657,604
Filed: September 8, 2000
Inventors: Jackson Brandenburg et al.
Sheet 14 of 19

48	Filter Engine returns response to Servlet.	RMI
49	Servlet returns response to SDK Web Server	Servlet
50	SDK Web Server returns response to Web Browser	HTTP

Fig. 13C

REPLACEMENT SHEET
Serial No.. 09/657,604
Filed: September 8, 2000
Inventors' Jackson Brandenburg et al.
Sheet 15 of 19



DSMS ACTIVE INTEGRATION

FIG. 14

REPLACEMENT SHEET
Serial No.: 09/657,604
Filed: September 8, 2000
Inventors: Jackson Brandenburg et al.
Sheet 16 of 19

#	Description	Protocol
1501 ~~1~~	User requests form that will require signing when submitted.	HTML UI
1502 ~~2~~	Web Browser sends request to Web Server.	HTTP
1503 ~~3~~	Web server forwards request to Web Application.	ISA
1504 ~~4~~	Web Application returns an HTML page for the web server to return which references the Client Interface	ISA
1505 ~~5~~	Web Server returns the HTML Page to Web Browser.	HTTP
1506 ~~6~~	Web Browser requests Client Interface from Web Server.	HTTP
1507 ~~7~~	Web Server retrieves Client Interface.	OS File System
1508 ~~8~~	Web Server returns Client Interface.	HTTP
1509 ~~9~~	User clicks the submit and sign button in the web page.	HTML UI
1510 ~~10~~	Web Browser calls Client Interface.	Client Interface Technology
1511 ~~11~~	Client Interface calls Windows PC/SC to have Smart Card sign data.	OS API
1512 ~~12~~	User enters PIN.	OS Dialog
1513 ~~13~~	Windows PC/SC calls Smart Card to sign data.	OS-Level Hardware Call
1514 ~~14~~	Windows PC/SC returns signed data to Client Interface	OS API
1515 ~~15~~	Client Interface returns signed data.	Client Interface Technology
1516 ~~16~~	Web Browser posts signed data.	HTTP
1517 ~~17~~	Web server passes signed posting to Web Application.	ISA
1518 ~~18~~	Integration Code added to the Web Application calls the Bank Interface to verify the signature on the form.	Bank Interface Technology
1519 ~~19~~	Bank Interface calls HSM OS Driver to sign request.	OS-API
1520 ~~20~~	HSM OS Driver calls HSM to sign request.	OS-Level Hardware Call

Fig 15A

1521 ~~21~~	HSM OS Driver returns signed request to Bank Interface	OS-API
1522 ~~22~~	Bank Interface calls the Relying Party's Bank.	Warranty/Status
1523 ~~23~~	Relying Party's Bank calls the Issuing Party's Bank.	Warranty/OCSP
1524 ~~24~~	Issuing Party's Bank returns a signed response to the Relying Party's Bank.	Warranty/OCSP
1525 ~~25~~	Relying Party's Bank calls the Root.	Warranty/OCSP
1526 ~~26~~	Root returns signed response to Relying Party's Bank	Warranty/OCSP
1527 ~~27~~	Relying Party's Bank returns signed response to the Bank	Warrant/Status
1528 ~~28~~	Bank Interface stores the signed data and the signed OK response from the relying party's bank into the Signed Documents repository.	Database-Access API
1529 ~~29~~	Bank Interface writes transaction log message.	File I/O
1530 ~~30~~	Bank Interface returns result to Web Application.	Bank Interface Technology
1531 ~~31~~	Web Application interprets the form post and returns the next page to the Web Server or an error.	ISA
1532 ~~32~~	Web Server returns the page to the Web Browser.	HTTP

T 15a